

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE  
ON APPEAL FROM THE EXAMINER TO THE BOARD  
OF PATENT APPEALS AND INTERFERENCES**

In re Application of:           Anders Vinberg  
Serial No.:                    10/091,065  
Filing Date:                 March 4, 2002  
Group Art Unit:              2452  
Examiner:                    Philip C. Lee  
Confirmation No.:            8010  
Title:                         METHOD AND APPARATUS FOR GENERATING CONTEXT-  
                                  DESCRIPTIVE MESSAGES

**MAIL STOP APPEAL BRIEF - PATENTS**

Commissioner for Patents  
P.O. Box 1450  
Alexandria, Virginia 22313-1450

Dear Sir:

**APPEAL BRIEF**

Appellant has appealed to the Board of Patent Appeals and Interferences ("*Board*") from the Final Office Action dated February 10, 2009 ("*Final Office Action*") and the Advisory Action dated April 21, 2009. Appellant filed a Notice of Appeal and Pre-Appeal Brief on May 8, 2009 with the statutory fee of \$540.00. This Appeal Brief is filed in response to Notice of Panel Decision from Pre-Appeal Brief Review dated July 30, 2009, finally rejecting Claims 1, 3-9, 11, 13-20, and 31-36.

**REAL PARTY IN INTEREST**

This Application is currently owned by Computer Associates Think, Inc. as indicated by:

an assignment recorded on 11/25/2002 from inventor Anders Vinberg to Computer Associates Think, Inc., in the Assignment Records of the PTO at Reel 013520, Frame 0080 (4 pages).

**RELATED APPEALS AND INTERFERENCES**

Appellant identifies the following appeal that may be related to or that may directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal:

Appn. Ser. No.:	10/091,067
Priority Info:	Claims the benefit of 08/892,919
Appellant:	Anders Vinberg
Represented by:	Baker Botts LLP
Assignee:	Computer Associates Think, Inc. (pursuant to assignment recorded at reel 013520, Frame 0528)
Status:	Appeal Brief to be submitted by Applicant on or before due August 30, 2009

Appellant additionally identifies the following appeal that may be related to or that may directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal:

Appeal No.:	2009-011165
Appn. Ser. No.:	09/949,101
Priority Info:	Claims the benefit of 08/892,919
Appellant:	Reuven (nmi) Battat, et al.
Represented by:	Baker Botts LLP
Assignee:	Computer Associates Think, Inc. (pursuant to assignment recorded at reel 012161, frame 0483)
Status:	Reply Brief submitted on December 30, 2008; awaiting decision of Board of Patent Appeals and Interferences

**STATUS OF CLAIMS**

Claims 1, 3-9, 11, 13-20, and 33-36 are pending and stand rejected pursuant to a Final Office Action dated February 10, 2009 ("*Final Office Action*") and a Notice of Panel Decision from Pre-Appeal Brief Review dated July 30, 2009 ("*Panel Decision*"). Specifically, the *Final Office Action* includes the following rejections:

1. Claims 1, 3-5, 9, 11, 13-15 and 33-36 are rejected under 35 U.S.C. § 103(a) as allegedly being unpatentable over U.S. Patent No. 6,125,390 issued to Touboul ("*Touboul*") and U.S. Patent No. 6,049,828 issued to Dev et al. ("*Dev*") in view of U.S. Patent No. 5,761,502 issued to Jacobs ("*Jacobs*").
2. Claims 6 and 16 are rejected under 35 U.S.C. § 103(a) as allegedly being unpatentable over *Touboul*, *Dev* and *Jacobs* in view of U.S. Patent No. 6,011,838 to Cox ("*Cox*").
3. Claims 7 and 17 are rejected under 35 U.S.C. § 103(a) as allegedly being unpatentable over *Touboul*, *Dev* and *Jacobs* in view of U.S. Patent No. 5,748,098 to Grace ("*Grace*").
4. Claims 8 and 18 are rejected under 35 U.S.C. § 103(a) as allegedly being unpatentable over *Touboul*, *Dev* and *Jacobs* in view of U.S. Patent No. 6,006,016 to Faigon et al. ("*Faigon*").
5. Claims 19-20 and 31-32 under 35 U.S.C. § 103(a) as allegedly being unpatentable over *Touboul*, *Dev* and *Jacobs* in view of U.S. Patent No. 5,933,601 to Fanshier ("*Fanshier*").

Claim 37 was added in the Response to Final Office Action submitted by Appellant on April 9, 2009 ("*Response to Final*"). However, the amendment has not been entered.

For the reasons discussed below, Appellant respectfully submits that the rejections of Claims 1, 3-9, 11, 13-20, and 33-36 are improper and should be reversed by the Board. Accordingly, Appellant presents Claims 11, 3-9, 11, 13-20, and 33-36 for Appeal. All pending claims are shown in Appendix A, attached hereto.

**STATUS OF AMENDMENTS**

In the Response to Final Office Action submitted by Appellant on April 9, 2009 (*“Response to Final”*), Appellant amended the claims by adding a new Claim 37 to depend from Claim 1. In the Advisory Action delivered on April 21, 2009 (*“Advisory Action”*), the Examiner refused to enter the amendment because such amendment would “raise new issues that would require further consideration and/or search.” (*Advisory Action*, pages 1-2). Claim 37 and its status as “Not Entered” is shown in Appendix A, attached hereto.

All other amendments submitted by Appellant have been entered by the Examiner.

**SUMMARY OF CLAIMED SUBJECT MATTER**

An exemplary IT enterprise is illustrated in Figure 1A. The IT enterprise 150 includes local area networks 155, 160 and 165. IT enterprise 150 further includes a variety of hardware and software components, such as workstations, printers, scanners, routers, operating systems, applications, and application platforms, for example. Each component of IT enterprise 150 may be monitored and managed in accordance with the present disclosure. (Page 4, lines 10-15.)

The various components of an exemplary management system 100 topology that can manage an IT enterprise in accordance with the present disclosure are shown in Figure 1B. The management system 100 includes at least one visualization workstation 105, an object repository 110, one or more management applications 115, and one or more management agents 120 associated with each management application 115. (Page 4, lines 16-20.)

The visualization workstation 105 provides a user access to various applications including a network management application 115. Workstation 105 interacts with an object repository 110 which stores and delivers requests, commands and event notifications. Workstation 105 requests information from object repository 110, sends commands to the object repository, and gets notification of events, such as status changes or object additions from it. The object repository 110 receives request information from the management application 115, which is fed by the management agents 120 responsible for monitoring and managing certain components or systems in an IT enterprise. (Page 4, lines 21-28.)

The management application 115 maintains object repository 110, in part, to keep track of the objects under consideration. The object repository 110 may be a persistent store to hold information about managed components or systems, such as a database. In an alternative embodiment, the management application 115 and object repository 110 may be integrated into a single unit such as a computer processing system that can hold information about managed components in, for example, a volatile memory and perform the tasks of the management application using, for example, the one or more processors (e.g., a management application processor) operable with logic encoded on a storage medium to execute the management applications. (Page 4, line 29 - page 5, line 4.)

As shown, one architectural aspect of the present system is that in normal operation, the visualization workstation 105 interacts primarily with the object repository 110. This reduces

network traffic, improves the performance of graphical rendering at the workstation, and reduces the need for interconnectivity between the visualization workstation 105 and a multitude of management applications 115, their subsystems and agents 120 existing in the IT enterprises. Of course, embodiments having other configurations of the illustrated components are contemplated, including a stand-alone embodiment in which the components comprise an integrated workstation. (Page 5, lines 5-12.)

In addition to handling requests, commands and notifications, object repository 110 may also handle objects describing the structure and operation of the management system 100. Such objects may describe the momentary state, load, and performance of the components and/or systems. Such objects may be populated using a manual process or an automatic discovery utility. (Page 5, lines 13-17.)

Referring now to Figure 2, components forming one embodiment of an alert system according to the present disclosure are shown. Management application 115 includes an alert system 200 for detecting and reporting alert conditions pertaining to managed components of the IT enterprise 150. The alert system 200 includes alert condition detection module 205 which oversees the status of system components by analyzing database 215, containing system objects that define the topology of the system. Through analysis of the system objects of database 215, alert condition detection module 205 may identify an actual or potential alert condition. Upon identifying an alert condition, module 205 generates an alert condition object and stores it in database 210. Alert notification module 220 periodically analyzes the alert condition objects of database 210, and reports relevant alert conditions represented by the objects. (Page 5, lines 18-28.)

Alert system 200 also includes an alert dialog manager 225 for generating messages that describe a context of a system object that is the subject of a reported alert condition of the managed system. In one embodiment, the context description may be provided as a result of one or more dialog requests received by alert dialog manager 225 from an operator, as illustrated by Figure 2. In an alternative embodiment, the alert notification module 220 and alert dialog manager 225 may be integrated, and the context description may be provided concurrently with an alert notification. (Page 5, line 29 - page 6, line 5.)

The context description of system object subject to an alert condition may include the physical location of the associated system component, the logical relationship of the system object to other system objects, the operating status of the system object, the business

process(es) associated with the system object, the interest/business groups associated with the system object. (Page 6, lines 6-10.)

Referring now to Figure 3, there is illustrated an exemplary flow diagram of methodology for reporting the context associated with an alert condition in accordance with one embodiment of the present disclosure. At block 305, an alert condition is detected. The alert condition may be an existing condition that requires operator attention, a warning regarding an existing condition or a predicted/potential condition that may require operator attention. Any technique known to those of skill in the art may be used in the detection of actual or potential alert conditions. (Page 6, lines 11-17.)

At block 310, an alert condition notification is generated. The notification may be embodied as text, motion video, audio or any other means for providing an alert. The alert condition notification may include an identification of the alert condition and/or a component of the system that is the subject of the alert. The alert condition notification is output to an operator at block 315. (Page 6, lines 18-22.)

At block 320, a determination is made whether a request to provide a description of the context of the alert has been received. If such a request has been received, the system continues processing at block 325. In an alternate embodiment, the system may be configured to automatically provide a complete or partial description of the context of the alert condition automatically, without requiring a request from an operator. In yet another alternate embodiment, the system may be configured to provide certain context information automatically, and certain other context information at the request of an operator. (Page 6, lines 23-30.)

At block 325, relevant system objects are analyzed to obtain context information. Which system objects that may be analyzed depend, in part, on the context information sought. For example, in order to provide the status of the component that is the subject of the alert condition, the system might analyze only the system object that represents the subject component. On the other hand, if the context request pertains to other components, such as for example, a request to list all components whose operation depend on the subject component, some or all of the system objects may be analyzed to determine their dependence on the subject component. (Page 7, lines 1-8.)



At block 330, a context message is generated describing the context of the alert condition and/or the subject component. The context message is output at block 335. (Page 7, lines 9-10.)

In the illustrated embodiment, blocks 320 through 335 may be performed more than once, allowing an operator to engage the system in a dialog. As an example, the system may output an alert notification at block 315 such as "There is a very high risk of a catastrophic slowdown in server uschdb02." (Page 7, lines 11-14.)

As in the present example, certain information may be replaced or rephrased before the alert notification is output. Such replacement of terms, which may also be applied to messages describing the context of the alert condition, may be performed in order to make such a message more natural and easier to understand by a human operator. In the present example, the system has replaced numeric quantifiers such as "75% risk" and "severity 4" with non-numeric quantifiers like "very high risk" and "catastrophic slowdown." (Page 7, lines 15-21.)

#### **Contextual Description of the Managed Object**

In order to identify the source of the problem, a user might request "what system is that?" seeking a more detailed contextual description of the managed component that is the subject of the alert notification. At block 335, the system may respond: (Page 7, lines 23-26.)

"uschdb02 is a mission-critical NT server in the Chicago web site server farm. It runs SQLServer. It has a replication server with automatic failover named uschdb02B, and this server is operational and in normal status." (Page 7, lines 27-29.)

Such a response identifies the context of the managed component in terms meaningful to the user. Elements of the message include: (Page 8, lines 1-2.)

uschdb02:                   The alert dialog manager 225 identifies the managed component in the sentence, to ensure that there is no misunderstanding and to make the sentence self-descriptive. (Page 8, lines 3-5.)

Mission-critical:       Database 215 maintains data describing the structure of the managed systems include an importance property for every object. The importance property may be defined at a class level or instance level, and may be propagated like status. The importance property is described in greater detail in the related commonly owned, co-pending, concurrently filed U.S. Patent

Application entitled "Method and Apparatus for Filtering Messages Based on Context" (Page 8, lines 6-13.)

NT server: Identifies the class of the relevant component. (Page 8, line 14.)

Chicago web site server farm: Identifies a grouping to which the relevant component belongs, which is discussed in greater detail below. (Page 8, lines 15-17.)

It runs SQLServer: This phrase identifies significant components contained in the managed component, in this example, a software system that runs on this server. In some cases, the function of a component may be carried out by a sub-component or subsystem hosted by the component. Since a component may host a number of sub-components and/or subsystems, in one embodiment only sub-components and/or subsystems having a threshold importance property may be reported to avoid/reduce confusion. (Page 8, lines 18-26.)

The final portion of the exemplary response, "it has a replication server with automatic failover named uschdb02B, and this server is operational and in normal status", provides other information about the managed object, that may be of interest to the operator. In this example, the system has information about a replication and failover configuration installed for the object, and describes it, with a reasonable amount of descriptive information about the replication server. The alert dialog manager 225 also provides the name and current status of the replication server. (Page 8, line 27 - page 9, line 3.)

#### **Identify the Topological Location of the Managed Object**

In order to identify the source of the problem, a user might request "where is the component located?" seeking a more detailed contextual description of the physical component that is the subject of the alert notification. At block 335, the system may respond: "uschdb02 is in Chicago, in the Headquarters building, in subnet xyz, in segment 1234." (Page 9, lines 5-10.)

The alert dialog manager 225 uses information about the location of the component in database 215 to determine the topological hierarchy related to the component, and creates a description based on a navigation down from the root of the hierarchy to the component. In the

present example, the system may respond: "uschdb02 is in Chicago, in HQ, in subnet xyz, in segment 1234." (Page 9, lines 11-15.)

### **Traffic Load Description**

Other information that an operator might wish to know to address an error condition includes a traffic load description. The operator may request "How busy is the component?", and the system might respond, for example, with "the traffic load on uschdb02 is high but within normal operating range.". Such a response illustrates how answers may be self-descriptive, to reduce the risk of misunderstandings over referents of pronouns. (Page 9, lines 17-23.)

### **Dependency Description**

In order to address some alert conditions, an operator may wish to identify dependency relationships between the component that is the subject of the alert condition and other components within the system. In order to facilitate providing such information, the alert dialog manager 225 supports dependency queries such as ""Who or what is dependent on the component?" (Page 9, lines 25-30.)

In response to the request for information, alert dialog manager 225 may reference database 215 at block 325 to analyze any dependency relationships associated with the subject component. The information regarding dependency relationships may be propagated up through a containment hierarchy. The alert dialog module 225 may generate and output a response, such as "All the web servers in the Chicago web site server farm are dependent on uschdb02.", for example, (Page 10, lines 1-6.)

The dependency relationships may be explicitly defined by a user or an application or they are deduced from discovered relationships. The dependency relationships may also be propagated to other components. For example, if an application depends on a database platform, a machine hosting the application also depends on the database platform. (Page 10, lines 7-11.)

In one embodiment, to make the context message more meaningful, the alert dialog manager 225 may avoid a long list of components in the initial message. Instead, at block 325, the alert dialog module 225 may identify a natural grouping of the components that can be used to generate a more meaningful description. For example, components may be identified as belonging to a pre-defined grouping with a distinct label. If database 215 already defines the

dependency relationship as pointing to a group, the alert dialog module 225 can readily create such a group-level description. If it does not, and the dependency relationships point to a number of components, the alert dialog module 225 can search for a natural grouping by listing all the groups that the components are members in, and analyzing the listing based on common definitions. (Page 10, lines 12-21.)

Examples of context messages resulting from such an analysis may include:

- 1) If there is a perfect match of the list of components with a group: "All the servers in the Chicago web site..."
  - 2) If some of the components in the list form a perfect match with a group: "All the servers in the Chicago web site plus the Detroit warehouse server..."
  - 3) If the components in the list match a group definition almost exactly: "All the servers in the Chicago web site except the SNA server..."
  - 4) If the components in the list form an imperfect match with a group: "Most of the servers in the Chicago web site..." or "Many of the servers in the Chicago web site..."
- (Page 10, line 22 - page 11, line 3.)

In one embodiment, the alert dialog module 225 compares available group definitions, and selects one with the best match as the basis for the description. If no useful grouping matches the list, the system may enumerate the systems individually if the list is short, or may neglect to specifically identify a specific dependency by using a phrase such as "several systems". To assist in the selection of a suitable grouping as the basis for a description, database 215 may include one or more indicators of the significance of different types of groupings. For example, membership in a business process such as Order Processing may be identified as more interesting, and therefore more useful as a descriptor, than the fact that servers are contained in a single network segment. Further, the alert dialog module 225 may support a request to explicit enumeration of dependencies, such as "the Chicago web site server farm includes uschap01, uschap02, uschap03, uschap05, uschap11, and uschap12". (Page 11, lines 5-16.)

In addition, the user may issue a query about the status of an entire group. In response to such a query, the system may generate a response that refers to the entire group, instead of listing each of the objects in the group. The following dialog illustrates such a group based status request: (Page 11, lines 17-20.)

"All the web servers in the Chicago web site server farm are dependent on uschdb02."

"What are their status?"

"The Chicago web site server farm is in normal status." (Page 11, lines 21-24.)

### **Selection of Relevant Information**

The analysis to obtain context information 325 is not limited to the objects of database 215. In some embodiments, alert dialog module 225 may utilize other information stores to obtain context information regarding the managed object. When an abundance of context information is obtained, it may be advantageous to present only a portion of the available information so as not to impair understanding of the large-scale situation. Accordingly, alert dialog module 225 may include control logic to determine which pieces of information to present. In one embodiment, alert dialog module 225 ranks each piece of information based on the importance ranking of each object, as well as predefined rules regarding what types of information are most interesting. These rules may be dependent on factors such as, for example, a component being managed or an operator identifier. (Page 11, line 26 - page 12, line 7.)

For example, when managing some networked computer systems, it may be more interesting to know what business process the system is a part of, rather than what network subnet it is a part of. The alert dialog module 225 may create the descriptive elements, and then rank them by relevance, including only the most important ones. (Page 12, lines 8-11.)

### **Impact Analysis**

In some embodiments, the object repository 110 stores data describing relationships among managed components, including, for example, containment relationships indicating which components are contained in another and various types of dependency relationships. Accordingly, the system may perform an impact analysis, which may be used to generate messages regarding all components affected by a diagnosed or predicted alert condition. (Page 12, lines 13-19.)

In one embodiment, the most important effects or problems may be reported to an operator. The management application 115 may employ logic to identify an impact analysis chain and create the alert notifications based on the most important object that is affected.

Since the importance property propagates along containment and dependency relationships, this is likely the highest object in the containment hierarchy. (Page 12, lines 20-24.)

### **Language Translation**

It is recognized that in a multinational system, operators may speak different native languages. Accordingly, in one embodiment the alert notification system includes translation capabilities. (Page 12, lines 27-30.)

Language translation may be performed in at least two ways: (1) a message may be generated in several languages, and one of the several languages may be selected for output to an operator, or (2) a message may be generated in some suitable language and translated in real time to another language for output to an operator. (Page 13, lines 1-4.)

Since complex systems may generate a wide variety of messages, messages that are constructed by intelligent subsystems in the form of complete sentences with context-dependent elements, it may not be practical to address translation of messages simply by manually translating the messages beforehand. Further, because the individual subsystems may be written in different countries and may run in different countries, it may not be realistic to enforce that all messages be generated in English. Therefore, according to one embodiment, the alert subsystem of management application 115 may generate messages in a predetermined language based on each subsystem, and the messages may be translated by industry-standard translation software. (Page 13, lines 5-13.)

This application is further related to U.S. Patent Nos. 5,958,012, 6,289,380 and 6,327,550, and co-pending U.S. Applications Serial Nos., 09/558,897, and 09/559,237, which are all incorporated in their entirety herein by reference. (Page 13, lines 14-16.)

Accordingly, it is to be understood that the drawings and description in this disclosure are proffered to facilitate comprehension of the system, and should not be construed to limit the scope thereof. It should be understood that various changes, substitutions and alterations can be made without departing from the spirit and scope of the system. (Page 13, lines 17-21.)

Claim 1 recites:

A method for reporting the context of an alert condition (e.g., Figure 3, reference numerals 305-335), comprising:

reporting an alert condition associated with a subject system object (e.g., Figure 3; reference numerals 305 and 310; Page 5, lines 18-28; Page 6, lines 11-22);

receiving, in response to the reporting of the alert condition, a user-generated text-based dialogue request specifying a user defined type of context data for the subject system object and one or more relevant system objects known to be associated with the subject system object (e.g., Figure 3; reference numerals 305 and 310; Page 5, lines 18-28; Page 6, lines 11-30; Page 7, line 23 through Page 9, line 3);

accessing a database to identify a group of system objects known to be associated with one another (e.g., Figure 1; reference numeral 110; Figure 2; reference numeral 215; Figure 3; reference numeral 325; Page 4, line 29 through Page 5, line 4; Page 5, lines 13-17; Page 5, lines 18-28; Page 7, lines 1-8);

identifying, from the group of system objects, a relevant system object that is known to be associated with the subject system object (e.g., Figure 3; reference numeral 325; page 7, lines 1-8; Page 10, line 12 through Page 11, line 24);

analyzing the subject system object associated with the alert condition and the relevant system object to obtain the context data (e.g., Figure 3; reference numeral 325; page 7, lines 1-8; Page 10, line 12 through Page 11, line 24; Page 11, line 26 through Page 12, line 24);

generating a context message based on the context data, the context message responsive to the user-generated request dialogue ; and

outputting the context message (e.g., Figure 3; reference numeral 335; page 7, lines 9-10).

Claim 8 recites:

A method for reporting the context of an alert condition (e.g., Figure 3, reference numerals 305-335), comprising:

reporting an alert condition associated with a subject system object (e.g., Figure 3; reference numerals 305 and 310; Page 5, lines 18-28; Page 6, lines 11-22);

receiving, in response to the reporting of the alert condition, a user-generated text-based dialogue request textually requesting context data for the subject system object and one or more relevant system objects known to be associated with the subject system object (e.g., Figure 3; reference numerals 305 and 310; Page 5, lines 18-28; Page 6, lines 11-30; Page 7, line 23 through Page 9, line 3);

accessing a database to identify a group of system objects known to be associated with one another (e.g., Figure 1; reference numeral 110; Figure 2; reference numeral 215; Figure 3; reference numeral 325; Page 4, line 29 through Page 5, line 4; Page 5, lines 13-17; Page 5, lines 18-28; Page 7, lines 1-8);

identifying, from the group of system objects, a relevant system object that is known to be associated with the subject system object (e.g., Figure 3;

reference numeral 325; page 7, lines 1-8; Page 10, line 12 through Page 11, line 24);

analyzing the subject system object associated with the alert condition and the relevant system object to obtain context data (e.g., Figure 3; reference numeral 325; page 7, lines 1-8; Page 10, line 12 through Page 11, line 24; Page 11, line 26 through Page 12, line 24);

generating a context message based on the context data, the context message responsive to the user-generated request dialogue (e.g., Figure 3; reference numeral 330; page 7, lines 9-10);

outputting the context message (e.g., Figure 3; reference numeral 335; page 7, lines 9-10); and

wherein generating includes replacing quantifiable context data with a qualitative identifier (e.g., Figure 3; reference numeral 330; page 7, lines 9-10 and 15-21; Page 11, line 27 through Page 12, line 11).

Claim 9 recites:

A system for reporting the context of an alert condition (e.g., Figures 1A, 1B, and 2, reference numerals 100 and 200; Page 4, line 10 through Page 6, line 5), comprising:

a management application processor (e.g., Figures 1B and 2, reference numeral 115; Page 4, line 29 through Page 5, line 4; Page 5, line 18 through Page 6, line 5) comprising:

means for reporting an alert condition associated with a subject system object (e.g., Figures 1B and 2; reference numerals 115 and 220; Figure 3; reference numerals 305 and 310; Page 5, line 18 through Page 6, line 22);

means for receiving, in response to the reporting of the alert condition, a user-generated text-based dialogue request specifying a user defined type of context data for the subject system object and one or more relevant system objects known to be associated with the subject system object (e.g., Figure 1B and 2, reference numeral 115; Figure 3; reference numerals 305 and 310; Page 5, lines 18 through Page 6, line 30; Page 7, line 23 through Page 9, line 3);

means for accessing a database to identify a group of system objects known to be associated with one another (e.g., Figures 1B and 2, reference numerals 110 and 115; Figure 3; reference numeral 325; Page 4, line 29 through Page 5, line 4; Page 5, lines 13-17; Page 5, lines 18-28; Page 7, lines 1-8);

means for identifying, from the group of system objects, a relevant system object that is known to be associated with the subject system object (e.g., Figures 1B and 2, reference numerals 110 and 115; Figure 3; reference numeral 325; Page 4, line 29 through Page 5, line 4; Page 5, lines 13-17; Page 5, lines 18-28; Page 7, lines 1-8; Page 10, line 12 through Page 11, line 24);

means for analyzing the subject system object associated with the alert condition and the relevant system object to obtain the context data (e.g., Figures 1B and 2, reference numerals 110 and 115; Figure 3; reference numeral 325; Page 4, line 29 through Page 5, line 4; Page 5, line 18 through



Page 6, line 5; Page 7, lines 1-8; Page 10, line 12 through Page 11, line 24; Page 11, line 26 through Page 12, line 24);

means for generating a context message based on the context data, the context message responsive to the user-generated request dialogue (e.g., Figures 1B and 2, reference numerals 110 and 115; Figure 3; reference numeral 330; Page 4, line 29 through Page 5, line 4; Page 5, line 18 through Page 6, line 5; Page 7, lines 9-10); and

means for outputting the context message (e.g., Figures 1B and 2, reference numerals 110 and 115; Figure 3; reference numeral 335; Page 4, line 29 through Page 5, line 4; Page 5, line 18 through Page 6, line 5; Page 7, lines 9-10).

Claim 11 recites:

Logic encoded in a storage medium (e.g., Figures 1A, 1B, and 2, reference numerals 100 and 200; Page 4, line 10 through Page 6, line 5; Page 11, line 27 through Page 12, line 7; Page 13, lines 513) and operable when executed to:

report an alert condition associated with a subject system object (e.g., Figure 3; reference numerals 305 and 310; Page 5, lines 18-28; Page 6, lines 11-22);

receive, in response to the reporting of the alert condition, a user-generated text-based dialogue request specifying a user defined type of context data for the subject system object and one or more relevant system objects known to be associated with the subject system object (e.g., Figure 3; reference numerals 305 and 310; Page 5, lines 18-28; Page 6, lines 11-30; Page 7, line 23 through Page 9, line 3);

access a database to identify a group of system objects known to be associated with one another (e.g., Figure 1; reference numeral 110; Figure 2; reference numeral 215; Figure 3; reference numeral 325; Page 4, line 29 through Page 5, line 4; Page 5, lines 13-17; Page 5, lines 18-28; Page 7, lines 1-8);

identify, from the group of system objects, a relevant system object that is known to be associated with the subject system object (e.g., Figure 3; reference numeral 325; page 7, lines 1-8; Page 10, line 12 through Page 11, line 24);

analyze the subject system object associated with the alert condition and the relevant system object to obtain the context data (e.g., Figure 3; reference numeral 325; page 7, lines 1-8; Page 10, line 12 through Page 11, line 24; Page 11, line 26 through Page 12, line 24);

generate a context message based on the context data, the context message responsive to the user-generated request dialogue (e.g., Figure 3; reference numeral 330; page 7, lines 9-10); and

output the context message (e.g., Figure 3; reference numeral 335; page 7, lines 9-10).

Dependent Claim 33 incorporates the elements of Claim 1, and further recites the following:

wherein the type of user defined context data is selected from the group consisting of location information for the subject system object, logical relationship information of the subject system object to other system objects, operational status information of the subject system object, or information regarding interest/business groups associated with the subject system object (e.g., Figures 1B and 2, reference numerals 110 and 115; Figure 3; reference numeral 325; Page 4, line 29 through Page 5, line 4; Page 5, line 18 through Page 6, line 5; Page 7, lines 1-8; Page 10, line 12 through Page 11, line 24; Page 11, line 26 through Page 12, line 24).

Dependent Claim 34 incorporates the elements of Claim 1, and further recites the following:

wherein the user-generated text-based dialogue request comprises a first user-generated text-based dialogue request specifying a user defined type of context data(e.g., Figure 3; reference numerals 305 and 310; Page 5, lines 18-28; Page 6, lines 11-30; Page 7, line 23 through Page 9, line 3); and further comprising:

after outputting the context message, receiving a second user-generated text-based dialogue request specifying a second user defined type of context data (e.g., Figure 3; reference numerals 305 and 310; Page 5, lines 18-28; Page 6, lines 11-30; Page 7, line 23 through Page 9, line 3; Page 7, lines 15-21).

Dependent Claim 35 incorporates the elements of Claim 1, and further recites the following:

wherein the user-generated text-based dialogue request textually requests the user defined type of context data (e.g., Figure 3; reference numerals 305 and 310; Page 5, lines 18-28; Page 6, lines 11-30; Page 7, line 23 through Page 9, line 3).

**GROUND OF REJECTION TO BE REVIEWED ON APPEAL**

Are Claims 1, 3-5, 9, 11, 13-15, and 36 unpatentable under 35 U.S.C. § 103(a) over U.S. Patent No. 6,125,390 issued to Touboul ("*Touboul*") and U.S. Patent No. 6,049,828 issued to Dev et al. ("*Dev*") in view of U.S. Patent No. 5,761,502 issued to Jacobs ("*Jacobs*")?

Is Claim 8 unpatentable under 35 U.S.C. § 103(a) over *Touboul*, *Dev* and *Jacobs* in view of U.S. Patent No. 6,006,016 to Faigon et al. ("*Faigon*")?

Are Claims 6 and 16 unpatentable under 35 U.S.C. § 103(a) over *Touboul*, *Dev* and *Jacobs* in view of U.S. Patent No. 6,011,838 to Cox ("*Cox*")?

Are Claims 7 and 17 unpatentable under 35 U.S.C. § 103(a) over *Touboul*, *Dev* and *Jacobs* in view of U.S. Patent No. 5,748,098 to Grace ("*Grace*")?

Is Claim 18 unpatentable under 35 U.S.C. § 103(a) over *Touboul*, *Dev* and *Jacobs* in view of U.S. Patent No. 6,006,016 to Faigon et al. ("*Faigon*")?

Are Claims 19-20 and 31-32 unpatentable under 35 U.S.C. § 103(a) over *Touboul*, *Dev* and *Jacobs* in view of U.S. Patent No. 5,933,601 to Fanshier ("*Fanshier*")?

Is Claim 33 unpatentable under 35 U.S.C. § 103(a) over *Touboul*, *Dev*, and *Jacobs*?

Is Claim 34 unpatentable under 35 U.S.C. § 103(a) over *Touboul*, *Dev*, and *Jacobs*?

Is Claim 35 unpatentable under 35 U.S.C. § 103(a) over *Touboul*, *Dev*, and *Jacobs*?

## ARGUMENTS

Claims 1, 3-9, 11 and 13-20, and 31-36 are pending and rejected. Claims 21-30 are withdrawn. As explained below, Appellant believes all claims to be allowable over the cited references. Accordingly, Appellant submits that these rejections are improper and should be reversed by the Board. Appellant addresses independent Claims 1, 8, 9, and 11 and dependent Claims 33-35 below.

### **I. The Legal Standard for Obviousness**

The question raised under 35 U.S.C. § 103 is whether the prior art taken as a whole would suggest the claimed invention taken as a whole to one of ordinary skill in the art at the time of the invention. One of the three basic criteria that must be established by an Examiner to establish a *prima facie* case of obviousness is that “the prior art reference (or references when combined) must teach or suggest ***all the claim limitations***.” See M.P.E.P. § 706.02(j) citing *In re Vaeck*, 947 F.2d 488, 20 U.S.P.Q.2d 1438 (Fed. Cir. 1991) (emphasis added). “***All words*** in a claim must be considered in judging the patentability of that claim against the prior art.” See M.P.E.P. § 2143.03 citing *In re Wilson*, 424 F.2d 1382, 1385 165 U.S.P.Q. 494, 496 (C.C.P.A. 1970) (emphasis added).

In addition, even if all elements of a claim are disclosed in various prior art references, which is certainly not the case here as discussed below, the claimed invention taken as a whole still cannot be said to be obvious without some reason why one of ordinary skill at the time of the invention would have been prompted to modify the teachings of a reference or combine the teachings of multiple references to arrive at the claimed invention.

The controlling case law, rules, and guidelines repeatedly warn against using an Appellant’s disclosure as a blueprint to reconstruct the claimed invention. For example, the M.P.E.P. states, “The tendency to resort to ‘hindsight’ based upon Appellant’s disclosure is often difficult to avoid due to the very nature of the examination process. However, impermissible hindsight must be avoided and the legal conclusion must be reached on the basis of the facts gleaned from the prior art.” M.P.E.P. § 2142.

The U.S. Supreme Court’s decision in *KSR Int’l Co. v. Teleflex, Inc.* reiterated the requirement that Examiners provide an explanation as to why the claimed invention would have been obvious. *KSR Int’l Co. v. Teleflex, Inc.*, 127 S.Ct. 1727 (2007). The analysis

regarding an apparent reason to combine the known elements in the fashion claimed in the patent at issue “should be made explicit.” *KSR*, 127 S.Ct. at 1740-41. “Rejections on obviousness grounds cannot be sustained by mere conclusory statements; instead, there must be some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness.” *Id.* at 1741 (internal quotations omitted).

The new examination guidelines issued by the PTO in response to the *KSR* decision further emphasize the importance of an explicit, articulated reason why the claimed invention is obvious. Those guidelines state, in part, that “[t]he key to supporting any rejection under 35 U.S.C. 103 is the clear articulation of the reason(s) why the claimed invention would have been obvious. The Supreme Court in *KSR* noted that the analysis supporting a rejection under 35 U.S.C. 103 should be made explicit.” *Examination Guidelines for Determining Obviousness Under 35 U.S.C. 103 in View of the Supreme Court Decision in KSR International Co. v. Teleflex Inc.*, 72 Fed. Reg. 57526, 57528-29 (Oct. 10, 2007) (internal citations omitted). The guidelines further describe a number of rationales that, in the PTO’s view, can support a finding of obviousness. *Id.* at 57529-34. The guidelines set forth a number of particular findings of fact that must be made and explained by the Examiner to support a finding of obviousness based on one of those rationales. *See id.*

**II. Claims 1, 3-5, 9, 11, 13-15, and 36 are allowable under 35 U.S.C. § 103(a) over the proposed *Touboul-Dev-Jacobs* combination**

Independent Claim 1, as presented on Appeal, recites:

A method for reporting the context of an alert condition, comprising:  
reporting an alert condition associated with a subject system object;  
receiving, in response to the reporting of the alert condition, a user-generated text-based dialogue request specifying a user defined type of context data for the subject system object and one or more relevant system objects known to be associated with the subject system object;  
accessing a database to identify a group of system objects known to be associated with one another;  
identifying, from the group of system objects, a relevant system object that is known to be associated with the subject system object;  
analyzing the subject system object associated with the alert condition and the relevant system object to obtain the context data;  
generating a context message based on the context data, the context message responsive to the user-generated request dialogue; and

outputting the context message.

Appellant respectfully submit that the cited references do not disclose, teach, or suggest the combination of elements recited above.

For example, the proposed *Touboul-Dev-Jacobs* combination does not disclose, teach, or suggest “receiving, in response to the reporting of the alert condition, a user-generated **text-based dialogue** request **specifying a user defined type of context data** for the subject system object,” as recited in Claim 1. In the *Final Office Action*, the Examiner acknowledges that *Touboul*, as the primary reference, does not disclose these limitations and instead relies upon *Dev*. (*Final Office Action*, page 3). Specifically, the Examiner points to a list of alarms displayed in Figure 10 of *Dev* and argues that the above-quoted limitations are taught by the act of “clicking on the condition red” to obtain more information (*Final Office Action*, page 3). However, Appellant respectfully points out that the alleged request of *Dev* (i.e., “clicking on the condition red”) does not allow (i) specification of the type of context data to be requested or (ii) the user to specify a user defined type of context data. Additionally, the clicking on a portion of text does not comprise “user-based text-based dialogue.”

As explained in one example scenario presented in Appellant’s Specification:

As an example, the system may output an alert notification at block 315 such as ‘There is a very high risk of a catastrophic slowdown in server uschdb02 . . . In order to identify the source of the problem, a user might request ‘what system is that?’ seeking a more detailed contextual description of the managed component that is the subject of the alert notification . . . [or] . . . [i]n order to identify the source of the problem, a user might request ‘where is the component located?’ seeking a more detailed contextual description of the physical component that is the subject of the alert notification.

(Specification, page 7, lines 12-26 and page 9, lines 6-7). By contrast, *Dev* makes it clear that no such specification by the user is possible. Rather, *Dev* merely discloses that by clicking on a particular alarm, the user may generically obtain “more information.” (*Dev*, col. 15, lines 16-18). The alleged request of *Dev* does not allow specification of the type of context data to be requested or the user to specify a user defined type of context data. In fact, *Dev* is completely devoid of any teaching that the alleged request “specif[ies] a user defined type of context data” as recited in Claim 1.

As stated above, the Examiner rejects the above-quoted limitations of Claim 1 by arguing that the system of *Dev* allegedly enables a user to define the severity of the event (e.g., “Condition Red”), and therefore, the act of clicking on “Condition Red” amounts to a user generated text-based dialogue specifying a user defined type of context data. (*Final Office Action*, page 14, lines 18 through page 15, line 1 (citing *Dev*, col. 8, lines 11-14). However, Appellant respectfully points out that this is a mischaracterization of *Dev*.

According to the portion of *Dev* relied on by the Examiner:

Event messages sent to the user interface can utilize a filter process that is specified by the user. The user can **specify model types and a minimum event severity for which events will be displayed** on the user screen. Events from unspecified model types or less than the minimum severity will not be displayed.

(*Dev*, col. 8, lines 11-14 (emphasis added)). That is, this passage of *Dev* merely discloses that a filter may be established by a user to limit the events displayed on the user’s screen. Thus, if the severity of the event is below a minimum level, the event will not be displayed. Even for those events that are displayed, the user may still only generically obtain “more information” by clicking on a particular alarm (i.e., “Condition Red”). (*Dev*, col. 15, lines 16-18). However, the clicking on a portion of text does not comprise “user-based text-based dialogue.” Further, “clicking on the condition red” by a user does not result in the specification of the type of context data to be requested or allow the user to specify a user defined type of context data.

For at least these reasons, Appellant submits that the act of “clicking on the condition red” as argued by the Examiner does not disclose, or even teach or suggest, “a user-generated text-based dialogue request specifying a user defined type of context data” recited in Claim 1. Consequently, Appellant respectfully contends that Claim 1 and each of its dependent claims (e.g., Claims 3-5 and 36) are in condition for allowance. For analogous reasons, Appellant further contends that Claims 9 and 11 and each of their dependent claims (e.g., Claims 13-15) are in condition for allowance.

### **III. Claim 8 is allowable under 35 U.S.C. § 103(a) over the proposed *Touboul-Dev-Jacobs* combination**

Independent Claim 8, as presented on Appeal, recites:

A method for reporting the context of an alert condition, comprising:

reporting an alert condition associated with a subject system object;  
receiving, in response to the reporting of the alert condition, a user-generated text-based dialogue request textually requesting context data for the subject system object and one or more relevant system objects known to be associated with the subject system object;  
accessing a database to identify a group of system objects known to be associated with one another;  
identifying, from the group of system objects, a relevant system object that is known to be associated with the subject system object;  
analyzing the subject system object associated with the alert condition and the relevant system object to obtain context data;  
generating a context message based on the context data, the context message responsive to the user-generated request dialogue;  
outputting the context message; and  
wherein generating includes replacing quantifiable context data with a qualitative identifier.

Appellant respectfully submit that the cited references do not disclose, teach, or suggest the combination of elements recited above.

For example, the proposed *Touboul-Dev-Jacobs* combination does not disclose, teach, or suggest “receiving, in response to the reporting of the alert condition, a user-generated **text-based dialogue** request **textually requesting** context data for the subject system object.” In the *Final Office Action*, the Examiner acknowledges that *Touboul*, as the primary reference, does not disclose these limitations and instead relies upon *Dev*. (*Final Office Action*, page 10). Again, the Examiner points to a list of alarms displayed in Figure 10 of *Dev* and argues that the above-quoted limitations are taught by the act of “clicking on the condition red” to obtain more information (*Final Office Action*, page 10). Specifically, to reject the recited limitations, the Examiner points to sections of *Dev* that describe a user being able to obtain “more information” on a particular alarm included in a list of alarms displayed in Figure 10 by “click[ing] on a particular alarm in the list of alarms.” (*Final Office Action*, page 10 (citing, *Dev* col. 15, lines 12-29; see also *Dev* col. 8, lines 31-37). More particularly, the Examiner argues, “since the request to obtain more information is generated by clicking on the text of the severity of ‘Condition Red,’ *Dev* teaches ‘textually requests context data.’” (*Final Office Action*, page 15). Appellant respectfully disagrees.

*Dev* merely discloses that by clicking on a particular alarm (i.e., “Condition Red”), the user may generically obtain “more information.” (*Dev*, col. 15, lines 16-18). Merely because the alleged request of *Dev* may be created by clicking on the textual words “Condition Red,” does disclose that the alleged request “textually request[s] context data.” The phrase



“Condition Red” does not textually request anything. Additionally, the phrase is not user-generated. Rather, the text is presented to the user for selection by the user.” Thus, the clicking on a portion of text does not comprise “user-generated text-based dialogue.” There is no dialogue received from the user in the system of *Dev*. For these reasons, the clicking on “Condition Red” is not a “user-generated text-based dialogue request textually requesting context data,” and it continues to be Appellant’s position that the act of “clicking on the text of the severity of ‘Condition Red’” as argued by the Examiner does not disclose, or even teach or suggest, “receiving, in response to the reporting of the alert condition, a user-generated text-based dialogue request textually requesting context data for the subject system object,” as recited in Claim 8.

For at least these reasons, Appellant respectfully contends that Claim 8 is in condition for allowance.

**IV. Claims 6 and 16 are allowable under 35 U.S.C. § 103(a) over the proposed *Touboul-Dev-Jacobs-Cox* combination**

Dependent Claims 6 and 16 depend on Claims 1 and 11, respectively. Accordingly, dependent Claims 6 and 16 are not obvious over the proposed *Touboul-Dev-Jacobs-Cox* combination at least because Claims 6 and 16 include the limitations of their respective independent claims, which Appellant has shown above to be allowable. Since Claims 6 and 16 incorporate the limitations of their respective independent claims, Appellant has not provided detailed arguments with respect to Claims 6 and 16. However, Appellant reserves the right to argue these claims if it becomes appropriate. Appellant respectfully requests reconsideration and allowance of Claims 6 and 16.

**V. Claims 7 and 17 are allowable under 35 U.S.C. § 103(a) over the proposed *Touboul-Dev-Jacobs-Grace* combination**

Dependent Claims 7 and 17 depend on Claims 1 and 11, respectively. Accordingly, dependent Claims 7 and 17 are not obvious over the proposed *Touboul-Dev-Jacobs-Grace* combination at least because Claims 7 and 17 include the limitations of their respective independent claims, which Appellant has shown above to be allowable. Since Claims 7 and 17 incorporate the limitations of their respective independent claims, Appellant has not

provided detailed arguments with respect to Claims 7 and 17. However, Appellant reserves the right to argue these claims if it becomes appropriate. Appellant respectfully requests reconsideration and allowance of Claims 7 and 17.

**VI. Claim 18 is allowable under 35 U.S.C. § 103(a) over the proposed *Touboul-Dev-Jacobs-Faigon* combination**

Dependent Claim 18 depends on Claim 11. Accordingly, dependent Claim 8 is not obvious over the proposed *Touboul-Dev-Jacobs-Faigon* combination at least because Claim 18 include the limitations of Claim 11, which Appellant has shown above to be allowable. Since Claim 18 incorporates the limitations of Claim 11, Appellant has not provided detailed arguments with respect to Claim 18. However, Appellant reserves the right to argue this claim if it becomes appropriate. Appellant respectfully requests reconsideration and allowance of Claim 18.

**VII. Claims 19-20 and 31-32 are allowable under 35 U.S.C. § 103(a) over the proposed *Touboul-Dev-Jacobs-Fanshier* combination**

Dependent Claims 19-20 and 31-32 depend on Claims 11 and 1, respectively. Accordingly, dependent Claims 19-20 and 31-32 are not obvious over the proposed *Touboul-Dev-Jacobs-Fanshier* combination at least because Claims 19-20 and 31-32 include the limitations of their respective independent claims, which Appellant has shown above to be allowable. Since Claims 19-20 and 31-32 incorporate the limitations of their respective independent claims, Appellant has not provided detailed arguments with respect to Claims 19-20 and 31-32. However, Appellant reserves the right to argue these claims if it becomes appropriate. Appellant respectfully requests reconsideration and allowance of Claims 19-20 and 31-32.

**VIII. Claim 33 is allowable under 35 U.S.C. § 103(a) over the proposed *Touboul-Dev-Jacobs* combination**

Claim 33 depends upon Claim 1 and includes the limitations of independent Claim 1. Accordingly, Claim 33 is not obvious over the proposed *Touboul-Dev-Jacobs* combination for the reasons discussed above with regard to Claim 1.

Additionally, Claim 33 recites claim elements that further distinguish the art. For example, Claim 33 recites that “the type of user defined context data is selected from the group consisting of location information for the subject system object, logical relationship information of the subject system object to other system objects, operational status information of the subject system object, or information regarding interest/business groups associated with the subject system object.” In the *Final Office Action*, the Examiner acknowledges that *Touboul, Dev, and Jacobs* “do not specifically teach user defined context data” selected from the group recited in Appellant’s claim. (*Final Office Action*, page 7). However, because *Dev* allegedly discloses user defined context data selected from any information contained in the event message, the Examiner maintains that the above quoted claim limitations are obvious over the teachings of *Dev*. (*Final Office Action*, page 7). Appellant disagrees.

First, *Dev* does not disclose user defined context data. Rather, *Dev* merely discloses that “[e]vent messages sent to the user interface can utilize a filter process that is specified by the user.” (*Dev*, col. 8, lines 11-12). Thus, a user can specify the types of objects for which the user will monitor and the severity that must be reached before a user is notified. (*Dev*, col. 8, lines 11-19). Although *Dev* discloses that “any information contained in the event message can be used for event filtering,” this only suggests that any information included in the alert message could be used for filtering out events that the user will not receive. Because such information in the alert messages and the messages themselves are system generated and not user-generated, *Dev* does not disclose, teach, or suggest user defined context data about an object. Accordingly, it would not have been obvious to modify *Dev* to include “the type of user defined context data is selected from the group consisting of location information for the subject system object, logical relationship information of the subject system object to other system objects, operational status information of the subject system object, or information regarding interest/business groups associated with the subject system object,” as recited in Appellant’s Claim 33.

For at least these reasons, Appellant respectfully contends that dependent Claim 33 is in condition for allowance.

**IX. Claim 34 is allowable under 35 U.S.C. § 103(a) over the proposed *Touboul-Dev-Jacobs* combination**

Claim 34 depends upon Claim 1 and includes the limitations of independent Claim 1. Accordingly, Claim 34 is not obvious over the proposed *Touboul-Dev-Jacobs* combination for the reasons discussed above with regard to Claim 1.

Additionally, Claim 34 recites claim elements that further distinguish the art. For example, Claim 34 recites that “after outputting the context message, receiving a second user-generated text-based dialogue request specifying a second user defined type of context data.” In the *Final Office Action*, the Examiner relies upon *Dev* for disclosure of the recited claim elements. (*Final Office Action*, page 8). Specifically, the examiner states that “clicking on other alarm[s]” discloses Appellant’s recited claim limitations. Appellant respectfully disagrees.

As stated above, *Dev* merely describes a user being able to obtain “more information” on a particular alarm included in a list of alarms displayed in Figure 10 by “click[ing] on a particular alarm in the list of alarms.” (*Final Office Action*, page 10 (citing, *Dev* col. 15, lines 16-18; col. 8, lines 11-14; Figure 10, reference numeral 420). As noted by the Examiner, *Dev* indicates that also “click on other alarms in the alarm list” to obtain “similar information . . . regarding other alarm conditions.” (*Dev*, col. 15, lines 27-29). However, clicking on the textual words “Condition Red” or another identifier of an alarm does disclose “user-generated text-based dialogue request specifying a second user defined type of context data,” as recited in Claim 34. To the contrary, the phrase “Condition Red” does not textually request anything. Additionally, the phrase is not user-generated. Rather, the text is presented to the user for selection by the user. Therefore, it continues to be Appellant’s position that the act of clicking on the text identifying an alarm does not disclose, or even teach or suggest “after outputting the context message, receiving a second user-generated text-based dialogue request specifying a second user defined type of context data,” as recited in Claim 34.

For at least these reasons, Appellant respectfully contends that dependent Claim 34 is in condition for allowance.

**X. Claim 35 is allowable under 35 U.S.C. § 103(a) over the proposed *Touboul-Dev-Jacobs* combination**

Claim 35 depends upon Claim 1 and includes the limitations of independent Claim 1. Accordingly, Claim 35 is not obvious over the proposed *Touboul-Dev-Jacobs* combination for the reasons discussed above with regard to Claim 1.

Additionally, Claim 35 recites claim elements that further distinguish the art. For example, Claim 35 recites that “the user-generated text-based dialogue request textually request the user defined type of context data.” In the *Final Office Action*, the Examiner relies upon *Dev* for disclosure of the recited claim elements. (*Final Office Action*, page 8). Again, the Examiner points to a list of alarms displayed in Figure 10 of *Dev* and argues that the above-quoted limitations are taught by the act of “clicking on the condition red” to obtain more information (*Final Office Action*, page 8). Specifically, to reject the recited limitations, the Examiner points to sections of *Dev* that describe a user being able to obtain “more information” on a particular alarm included in a list of alarms displayed in Figure 10 by “click[ing] on a particular alarm in the list of alarms.” (*Final Office Action*, page 10 (citing, *Dev* col. 15, lines 16-18; col. 8, lines 11-14; Figure 10, reference numeral 420). More particularly, the Examiner argues, “since the request to obtain more information is generated by clicking on the text of the severity of ‘Condition Red,’ *Dev* teaches ‘textually requests context data.’” (*Final Office Action*, page 15). Appellant respectfully disagrees.

As stated above with regard to Claim 8, *Dev* merely discloses that by clicking on a particular alarm (i.e., “Condition Red”), the user may generically obtain “more information.” (*Dev*, col. 15, lines 16-18). Merely because the alleged request of *Dev* may be created by clicking on the textual words “Condition Red,” does disclose that the alleged request “textually request[s] context data.” The phrase “Condition Red” does not textually request anything. Additionally, the phrase is not user-generated. Rather, the text is presented to the user for selection by the user.” Certainly, the clicking on “Condition Red” is not a “user-generated text-based dialogue request textually requesting context data.” Therefore, it continues to be Appellant’s position that the act of “clicking on the text of the severity of ‘Condition Red’” as argued by the Examiner does not disclose, or even teach or suggest that “the user-generated text-based dialogue request textually request the user defined type of context data,” as recited in Claim 35.

For at least these reasons, Appellant respectfully contends that dependent Claim 35 is in condition for allowance.

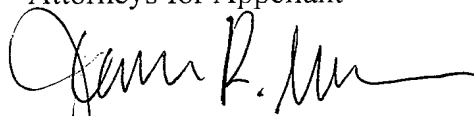
**CONCLUSION**

Appellant has demonstrated that the present invention, as claimed, is clearly distinguishable over the prior art cited by the Examiner. Therefore, Appellant respectfully requests the Board to reverse the final rejections and instruct the Examiner to issue a Notice of Allowance with respect to all pending claims.

The Commissioner is hereby authorized to charge \$540.00 for filing this Brief in support of an Appeal to Deposit Account No. 02-0384 of Baker Botts, L.L.P. No other fees are believed due; however, the Commissioner is authorized to charge any additional fees or credits to Deposit Account No. 02-0384 of Baker Botts, L.L.P.

Respectfully submitted,

BAKER BOTTS L.L.P.  
Attorneys for Appellant



Jenni R. Moen  
Reg. No. 52,038  
(214) 415-4820

Dated: August 25, 2009

**Correspondence Address:**

at Customer No. **05073**

**APPENDIX A**

*Pending Claims*



1. **(Rejected)** A method for reporting the context of an alert condition, comprising:

reporting an alert condition associated with a subject system object;

receiving, in response to the reporting of the alert condition, a user-generated text-based dialogue request specifying a user defined type of context data for the subject system object and one or more relevant system objects known to be associated with the subject system object;

accessing a database to identify a group of system objects known to be associated with one another;

identifying, from the group of system objects, a relevant system object that is known to be associated with the subject system object;

analyzing the subject system object associated with the alert condition and the relevant system object to obtain the context data;

generating a context message based on the context data, the context message responsive to the user-generated request dialogue; and

outputting the context message.

2. **(Canceled)**

3. **(Rejected)** The method of claim 1, wherein the analyzing includes determining properties of the subject system object.

4. **(Rejected)** The method of claim 1, wherein analyzing includes determining a physical location of a component represented by the subject system object.

5. **(Rejected)** The method of claim 1, wherein analyzing includes determining a logical relationship of a component represented by the subject system object to a component represented by the relevant system object.

6. **(Rejected)** The method of claim 1, wherein analyzing includes determining a traffic load associated with the subject system object.

7. **(Rejected)** The method of claim 1, wherein the relevant system object representing a component that is dependent on a component represented by the subject system object.

8. **(Rejected)** A method for reporting the context of an alert condition, comprising:

reporting an alert condition associated with a subject system object;

receiving, in response to the reporting of the alert condition, a user-generated text-based dialogue request textually requesting context data for the subject system object and one or more relevant system objects known to be associated with the subject system object;

accessing a database to identify a group of system objects known to be associated with one another;

identifying, from the group of system objects, a relevant system object that is known to be associated with the subject system object;

analyzing the subject system object associated with the alert condition and the relevant system object to obtain context data;

generating a context message based on the context data, the context message responsive to the user-generated request dialogue;

outputting the context message; and

wherein generating includes replacing quantifiable context data with a qualitative identifier.

9. **(Rejected)** A system for reporting the context of an alert condition, comprising:  
a management application processor comprising:

means for reporting an alert condition associated with a subject system object;

means for receiving, in response to the reporting of the alert condition, a user-generated text-based dialogue request specifying a user defined type of context data for the subject system object and one or more relevant system objects known to be associated with the subject system object;

means for accessing a database to identify a group of system objects known to be associated with one another;

means for identifying, from the group of system objects, a relevant system object that is known to be associated with the subject system object;

means for analyzing the subject system object associated with the alert condition and the relevant system object to obtain the context data;

means for generating a context message based on the context data, the context message responsive to the user-generated request dialogue; and

means for outputting the context message.

10. **(Canceled)**

11. **(Rejected)** Logic encoded in a storage medium and operable when executed to:  
report an alert condition associated with a subject system object;

receive, in response to the reporting of the alert condition, a user-generated text-based dialogue request specifying a user defined type of context data for the subject system object and one or more relevant system objects known to be associated with the subject system object;

access a database to identify a group of system objects known to be associated with one another;

identify, from the group of system objects, a relevant system object that is known to be associated with the subject system object;

analyze the subject system object associated with the alert condition and the relevant system object to obtain the context data;

generate a context message based on the context data, the context message responsive to the user-generated request dialogue; and

output the context message.

12. **(Canceled)**

13. **(Rejected)** The logic of claim 11, wherein when analyzing at least the subject system object, the logic is further operable to determine properties of the subject system object.

14. **(Rejected)** The logic of claim 11, wherein when analyzing at least the subject system object, the logic is further operable to determine a physical location of a component represented by the subject system object.

15. **(Rejected)** The logic of claim 11, wherein when analyzing at least the subject system object, the logic is further operable to determine a logical relationship of a component represented by the subject system object to a component represented by the relevant system object.

16. **(Rejected)** The logic of claim 11, wherein when analyzing at least the subject system object, the logic is further operable to determine a traffic load associated with the subject system object.

17. **(Rejected)** The logic of claim 11, wherein the relevant system object representing a component that is dependent on a component represented by the subject system object.

18. **(Rejected)** The logic of claim 11, wherein when generating the logic is further operable to replace quantifiable user defined context data with a qualitative identifier.

19. **(Rejected)** The logic of claim 11, wherein the relevant system object represents a component that is a sub-component of a component represented by the subject system object.

20. **(Rejected)** The logic of claim 11, wherein the relevant system object represents a component that is in a grouping with a component represented by the subject system object.

21. **(Withdrawn)** A system for reporting the context of an alert condition, comprising:

a database storing data associated with a plurality of system objects, the plurality of objects comprising at least a subject system object and a relevant object;

a management application module coupled to the database and operable to:

report an alert condition associated with a subject system object;

identify a relevant system object that is associated with the subject system object;

analyze the subject system object associated with the alert condition and the relevant system object to obtain context data;

generate a context message based on the context data; and

output the context message.

22. **(Withdrawn)** The system of claim 21, wherein the management application is further operable to receive a request to report the context of the alert condition.

23. **(Withdrawn)** The system of claim 21, wherein when analyzing at least the subject system object, the management application is operable to determine properties of the subject system object.

24. **(Withdrawn)** The system of claim 21, wherein when analyzing at least the subject system object, the management application is operable to determine a physical location of a component represented by the subject system object.

25. **(Withdrawn)** The system of claim 21, wherein when analyzing at least the subject system object, the management application is operable to determine a logical relationship of a component represented by the subject system object to a component represented by the relevant system object.

26. **(Withdrawn)** The system of claim 21, wherein when analyzing at least the subject system object, the management application is operable to determine a traffic load associated with the subject system object.

27. **(Withdrawn)** The system of claim 21, wherein the relevant system object represents a component that is dependent on a component represented by the subject system object.

28. **(Withdrawn)** The system of claim 21, wherein when generating the context message, the management application is operable to replace quantifiable context data with a qualitative identifier.

29. **(Withdrawn)** The system of claim 21, wherein the relevant system object represents a component that is a sub-component of a component represented by the subject system object.

30. **(Withdrawn)** The system of claim 21, wherein the relevant system object represents a component that is in a grouping with a component represented by the subject system object.

31. **(Rejected)** The method of claim 1, wherein the relevant system object represents a component that is a sub-component of a component represented by the subject system object.

32. **(Rejected)** The method of claim 1, wherein the relevant system object represents a component that is in a grouping with a component represented by the subject system object.

33. **(Rejected)** The method of claim 1, wherein the type of user defined context data is selected from the group consisting of location information for the subject system object, logical relationship information of the subject system object to other system objects, operational status information of the subject system object, or information regarding interest/business groups associated with the subject system object.

34. **(Rejected)** The method of claim 1, wherein the user-generated text-based dialogue request comprises a first user-generated text-based dialogue request specifying a user defined type of context data; and further comprising:

after outputting the context message, receiving a second user-generated text-based dialogue request specifying a second user defined type of context data.

35. **(Rejected)** The method of claim 1, wherein the user-generated text-based dialogue request textually requests the user defined type of context data.

36. **(Rejected)** The method of claim 1, wherein the context message contains the user defined type of context data specified in the request.

37. **(Not Entered)** The method of claim 1, wherein the alert condition is reported to a user; and further comprising:

enabling the user to specify the user defined type of context data after receiving the alert condition.

**APPENDIX B**

***Evidence Appendix***

Other than the references attached to this Appeal Brief as Appendices A and B, no evidence was submitted pursuant to 37 C.F.R. §§ 1.130, 1.131, or 1.132, and no other evidence was entered by the Examiner and relied upon by Appellant in the Appeal.



**APPENDIX C**

***Related Proceedings Appendix***

As stated on Page 3 of this Appeal Brief, to the knowledge of Appellant's Counsel, there are no known appeals, interferences, or judicial proceedings that will directly affect or be directly affected by or have a bearing on the Board's decision regarding this Appeal.